Magic Window™

word processing system

artsci inc

Magic Window

Window

word processing system

artsci inc

THE MAGIC WINDOW

Word Processing System

Created by Gary Shannon

Revised and Documented by Bill Depew

For APPLE II and APPLE II PLUS

TABLE OF CONTENTS

The MAGIC WINDOW is a word processing system. As with other systems, it allows text to be entered, edited, printed, and saved to be restored from diskette. Although similar in function, the MAGIC WINDOW is unique in the way it performs these functions. Most systems seperate the tasks of editing text and formatting for printing. This seperation requires formatting commands to be placed in the text file. The MAGIC WINDOW has no need for these commands because it formats the text as you enter it. What you see is truly what you get. There is no guesswork in laying out your document.

The simplest and best understood word processing machine is the standard typewriter. We have tried to retain the simple operating features of the typewriter while providing complete editing facilities. Because of the similarities, the operation of the MAGIC WINDOW can easily be described in terms of a typewriter.

Before typing, paper is required. In the MAGIC WINDOW, "paper" is inserted by defining the form size and margin locations in the FORMAT SUBSYSTEM. After these are established (the paper is installed), you start typing. The cursor, like a typewriter's keys, does not move, but remains in the center of the video screen as the text passes by, like the carriage and paper on the typewriter.

The APPLE II video output displays 40 characters on a line. Generally, the format, simulating paper, will be wider than this. An editing "window" allows you to view an area of 20 characters on each side of the cursor. As you type, the text moves through the window, allowing you to see the area of text around the cursor, but not the entire line. There is a similar restriction in how many lines can be seen at one time. Tab stops and cursor motion commands accommodate the window concept and make reviewing the text as easy as possible. The most important result of this approach

is the lack of abstraction normally associated with CRT editing systems. You know what the document will look like, and where the pages end, before you print it.

File management, printing, and system configuration are controlled by menu driven SUBSYSTEMS. This avoids complicated command sequences and syntax. Function selection is simple and syntax free.

The system is designed to boot up already configured as you like it. This is accomplished by saving the system environment on your data diskette. You boot the system from the master diskette, remove it, then boot your data diskette. Individual files also retain formatting information, restoring the system to the state it was in when the file was saved.

The printer interface is flexible and will handle ROM driven "intelligent interface" cards and RAM resident drivers. The files are stored on standard DOS 3.2 initialized diskettes permitting easy access to your text files for custom applications. In general, care has been taken to avoid restrictions in configuration or application.

We believe the MAGIC WINDOW is one of the easiest systems to learn to use. We hope you will agree and use this tool to lessen the effort of communication and to improve your productivity.

NOTATION

Throughout this manual, keys on the APPLE keyboard will be identified by upper case letters. The sequence SHIFT/CTRL-P means to hold the SHIFT and CTRL keys depressed, then press P.

SYSTEM HARDWARE REQUIREMENTS

The following is the minimum hardware configuration needed to use the MAGIC WINDOW.

1. APPLE II or APPLE II+ computer (w/display).
2. DISK II and controller card.
3. 32 kilobytes of RAM (48K recommended).
4. Printer and interface card.

The following optional equipment is directly supported and will enhance system performance.

5. Hardware Lower Case Adapter (such as Paymar's)
6. APPLE LANGUAGE SYSTEM (10K more text storage)

Many applications can be realized with user written custom support programs that use MAGIC WINDOW files. These include but are not limited to:

Electronic Mail - MODEM required

Softcopy (as opposed to hardcopy) files generated by the MAGIC WINDOW can be transfered via a MODEM and BASIC control program to be printed at the receiving site.

Customized Form Letters

Form letters may be created as softcopy with field markers. These markers can be replaced with data from the data base when the file is printed.

Data Collection

The MAGIC WINDOW page (or record) size is variable from 1 line to 84 lines. The record (page) boundaries are clearly shown by the editor making it ideal for data entry. Just think field instead of line and record instead of page.

These are just a few of the many possible ways to use the MAGIC WINDOW.

BOOT CAMP

BOOTING SYSTEM MASTER DISKETTE

To bring the MAGIC WINDOW into memory for use, boot the system master diskette. The system diskette was designed to boot on any APPLE II or APPLE II+ computer system. It will boot on LANGUAGE CARD equipped systems without previously booting the BASICS disk supplied with those systems.

To boot, insert diskette then type: (n=disk slot)

> PR#n RETURN from either BASIC
> or
> nCTRL-P RETURN to boot from the system monitor.

During the boot, the system is checked for a LANGUAGE CARD. If one is installed, the MAGIC WINDOW is loaded into the card. If the card is not available, the program loads into on board RAM. Users with LANGUAGE SYSTEMS will gain about 10k bytes of text storage above that of users without. Once in memory the system executes, displaying the title, authors, and copyright notice.

DATA DISKETTES

You are now instructed to:

> INSERT DATA DISKETTE THEN PRESS RETURN
>
> PRESS ESC IF DISKETTE NOT AVAILABLE

CONFIGURATION BOOT FILE

As well as storing your files, the data diskette can preserve your operating environment. Any floppy diskette initialized for APPLE DOS 3.2 is suitable for data storage. It can also have a file named SYS.OPTIONS. This file contains information about your system. Pressing RETURN will boot the file and configure the system. If you have more than one drive on line you may also specify the unit by typing ,Ss,Dd,Vv RETURN. These terms are explained in your DOS manual. Note the leading comma, because

it is required. If this is your first time through you won't have the necessary file available so press ESC to skip the CONFIGURATION BOOT.

SUBSYSTEM MENU

        EDITOR SUBSYSTEM
        FORMAT SUBSYSTEM
        FILER SUBSYSTEM
        PRINTER SUBSYSTEM
        CONFIGURATION SUBSYSTEM
        EXIT TO BASIC

        The SUBSYSTEM MENU is the vehicle for entering, exiting, and transferring control within the MAGIC WINDOW. The menu offers six functions for you to select from. Where ever you go from the MENU, you will return to it before taking another path.

MENU SELECTION

        To select a function move the white bar, or cursor, with the ARROW keys until it covers the function you want. Press RETURN to execute that function. All of the subsystems use this common selection method. The CONFIGURATION SUBSYSTEM is a good place to start. Position the cursor, press RETURN and away you go.

CONFIGURATION SUBSYSTEM

        LOWER CASE VIDEO...........NO
        UPPER CASE ONLY............NO
        KEYPRESSED SOUND...........YES
        PRINTER SLOT...............1
        RAM PRINTER DRIVER.........NO
        AUTO LINE FEED.............YES
        PRINT MSB HIGH.............YES

        SUBSYSTEM MENU
        CHANGE CONFIGURATION OPTIONS
        UPDATE BOOT CONFIGURATION FILE
        BOOT CONFIGURATION FILE
        LOAD RAM PRINTER DRIVER

CONFIGURATION OPTIONS

The preferences declared at the top are the subsystem options. Of course, there is a subsystem function that allows you to change the options. When CHANGE OPTIONS is executed, a second bar will appear on the first option. Select the option to be changed the same way you have selected the function to be executed. There are two types of options; logical and numeric. When the white bar is positioned on a logical option, pressing RETURN will change a YES to NO and vice versa. If the option is numeric, a prompt line appears indicating the minimum and maximum values the system will accept for that option. Enter the desired value and press RETURN. After changing options, press ESC to return to function selection. So much for technique, let's set up your system.

LOWER CASE VIDEO

If you have a hardware lowercase character generator, such as the Lower Case Adapter offered by Dan Paymar, say YES and use it! If not you might consider buying one. They are inexpensive (<$75) and worth every penny. This option tells the editor how to display characters on the video screen. If LOWER CASE VIDEO is selected, true lower case ASCII will be placed on the screen. If only upper case is available there is a method for identifying the case. Upper case characters are displayed as inverted (black on white) while lower case characters are normal.

UPPER CASE ONLY

Sometimes your application may not require or even forbid lower case characters. Editing BASIC programs is one example. When selected, UPPER CASE ONLY forces the shift status to capitals, and does not allow lower case charaters to be entered. With only upper case characters in the file, there is no need to invert them for clarity.

KEYPRESSED SOUND

When selected, the speaker will tick when you press a key while editing text. Like a typewriter, this audio feedback gives a positive indication that a key was pressed. For those who like quiet, so be it.

PRINTER SLOT

When you use an "intellegent printer interface" such as those manufactered by APPLE COMPUTER INC., this option identifies the slot where the card will be found. This is a numeric type option. With the white bar positioned over a numeric option, pressing RETURN will cause a prompt line to be printed. This prompt will tell you the minimum and maximum values the system will accept for that option and ask you to enter a value. Enter the slot number of the card then press RETURN.

RAM PRINTER DRIVER

For users with custom printer interfaces without ROM firmware the option of printing to a RAM based driver routine is provided. When selected, this option causes a file called PRINTER to be loaded from the data diskette at boot time. The file may also be booted by a CONFIGURATION function. Information about designing RAM based drivers can be found in APPENDIX A of this manual. If you have a smart printer card answer NO.

AUTO LINE FEED

When selected, a line feed character will be printed after each carriage return. Some printers need the line feed while others supply it automatically. Consult your printer manual to see if this is required.

PRINT MSB HIGH

This controls the state of the Most Significant Bit of every byte sent to the printer. YES will force it high. Consult your printer manual to see if this bit should be low.

When you are finished configuring your system, press ESC. ESC exits the change option mode and returns you to function selection.

UPDATE BOOT CONFIGURATION

If you have an initialized diskette available you may want to update the boot configuration file so you won't have to do this every time you boot. Before executing this function, place an initialized data diskette into the current drive. Any DOS 3.2 initialized diskette will work as a data diskette. To initialize a blank diskette, follow the instructions provided in the DOS manual. Executing this command will write the configuration options onto the disk into a file called SYS.OPTIONS. When you boot, inserting that diskette when prompted will restore your configuration options.

Although your system may not change much, the tasks it is to perform come in all shapes and sizes. Sizing up the job is handled by the FORMAT SUBSYSTEM. Return to the SUBSYSTEM MENU (first function) then enter the FORMAT SUBSYSTEM.

FORMAT SUBSYSTEM

        PAGE LENGTH.............66
        TOP MARGIN..............6
        TEXT LENGTH............54
        PAGE WIDTH.............80
        LEFT MARGIN............8
        TEXT WIDTH.............64
        DOUBLE SPACE...........NO

        SUBSYSTEM MENU
        CHANGE FORMAT OPTIONS

The FORMAT SUBSYSTEM is used to define the text boundaries and form size you will be using. Page formatting is normally done after the file is cleared (FILER SUBSYSTEM) and before text is entered.

PAGE LENGTH & PAGE WIDTH

These options define the form size measured in lines per page and characters per line. Standard vertical spacing is six lines per inch. Using 10 pica character spacing, 85 characters fit from edge to edge on an 8 1/2 by 11 inch sheet.

TOP MARGIN & LEFT MARGIN

The MARGIN values define the number of margin lines and characters before the edges of the text area.

TEXT LENGTH & TEXT WIDTH

The WIDTH values define how many lines and characters of text will appear after the margins. The maximum TEXT WIDTH is 80 characters when using the MAGIC WINDOW.

The numbering starts at 1. Line 1, character 1 is at the top left edge of the paper. The bottom and right margins are calculated by adding the margin and text values. If you enter values that are not possible, more lines of text than the page length for example, an error message will appear when you try to exit the subsystem. You will not be able to exit until the values are legitimate.

DOUBLE SPACING.

When selected, DOUBLE SPACING is used by the editor to make the job a little easier. Carriage RETURN will cause two line feeds. Overtyping a line causes an extra line to be inserted for you and SPLIT and GLUE work with double spacing in mind.

Text can still be entered on every line, but you will have to use the cursor up and down functions to get between the lines. When using DOUBLE SPACING the TEXT LENGTH value should be an even number.

PAGE FORMAT LAYOUT
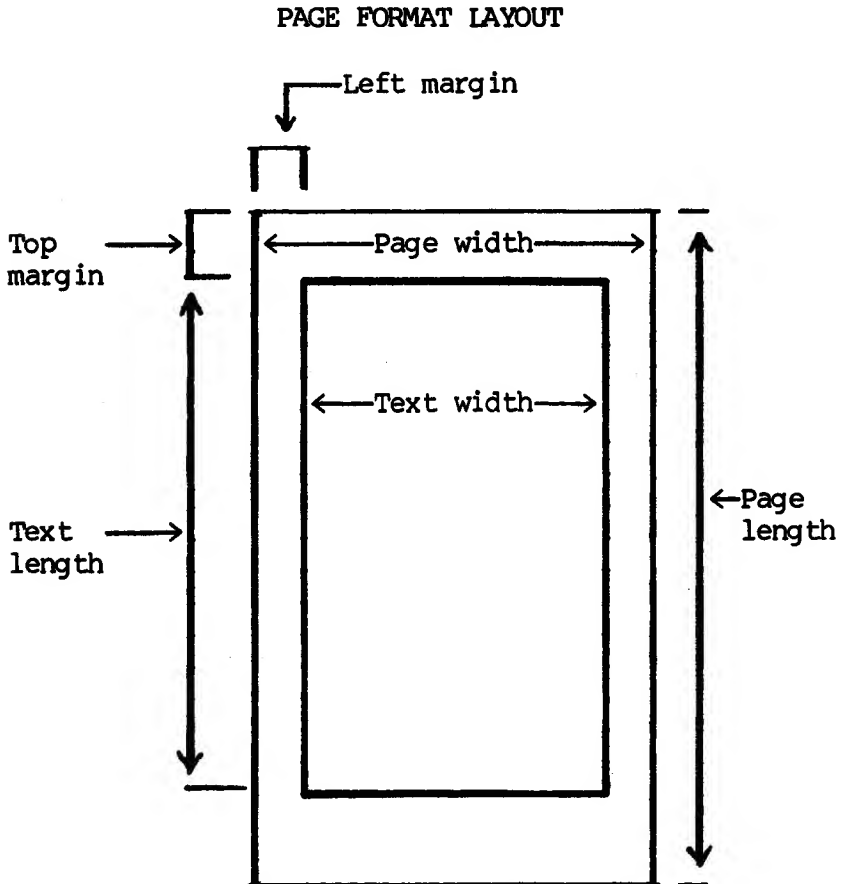
Once again return to the SUBSYSTEM MENU. Every thing is ready and we can now start typing. To enter and edit text execute the EDITOR SUBSYSTEM.

THE EDITOR DISPLAY

When editing files, you will be looking at the editor display. It consists of one line of positional information and 23 lines of text "window" display. In the center of the window is located the cursor. The cursor appears as a flashing underscore. The positional information at the top of the screen locates the cursor on the paper. The first entry is the line number, appearing as L followed by a number. This is the vertical position of the cursor on the page. The second item is character number indicated ·by the underscore character followed by a number. This is the horizontal position of the cursor. The last value is the page number shown as P followed by the page value. All three values represent absolute position starting at one. The MAGIC WINDOW text file can be considered a series of pages connected at top and bottom. You are free to roam through these pages using the cursor positioning commands. The travel of the cursor is restricted to the text areas defined in the FORMAT SUBSYSTEM.

THE CURSOR

The position of the cursor shows where the next character typed will go in the file. The appearance of the cursor reflects the "shift key" mode. If it appears as a white underscore on a black background, the next character typed will be put into the file as a lower case character. When the cursor is inverted as black against white, the next character will be upper case. (Note that this has nothing to do with the SHIFT key on the Apple II keyboard.) The display of the cursor alternates with the display of the character that is at that position in the file.

CASE SELECTION

There are three modes of case control. They are lower case, upper case, and upper case lock. Case mode is selected with the ESC key. The resulting mode depends on the mode in effect before ESC is

pressed. If in lower case, the ESC key selects upper case. If in upper case, upper case lock is selected. Pressing ESC while in upper case lock mode returns us to lower case. As long as upper case lock is not selected, entering any character into the file returns us to lower case mode.

ENTERING TEXT

This is straightforward. Pressing a key causes a character to be placed in the file at the cursor position. The cursor position is then updated for the next character. Use the ESC key to select the desired case as described above. When you approach the end of the line and overtype, the word will be shifted onto the next line. The new line is blank and was inserted automatically for you. Carriage RETURN performs its standard function. When you reach the bottom margin the cursor will skip over it to the first line of the next page. You are never allowed to type in the margins. You are permitted a title line placed in a margin as will be discussed in the EDITING COMMANDS section.

EDITING COMMANDS

Editing commands are issued using the CTRL key. Holding the CTRL key depressed and pressing a character key will cause a command to be executed. The next section of the manual is devoted to these functions and their use. You will have plenty of time to experiment using that section for reference. For the sake of speed let's assume you have already typed and edited a file and want to do something with it.

SUBSYSTEM MENU (exit EDITOR)

Command Key => SHIFT/CTRL-P (CTRL-@)

Function: EXIT TO SUBSYSTEM MENU

To exit the editor at any time, hold both the CTRL and SHIFT keys depressed then press P.

FILER SUBSYSTEM

        DISK SLOT..................6
        DISK DRIVE.................1
        DISK VOLUME................0

        SUBSYSTEM MENU
        CHANGE DISK OPTIONS
        FILES DIRECTORY
        SAVE FORMATTED FILE
        LOAD FORMATTED FILE
        UNFORMATTED LOAD/SAVE
        DELETE FILE
        CLEAR FILE IN MEMORY

        File maintenance is performed by the FILER
SUBSYTEM. The MAGIC WINDOW uses floppy diskettes
for data storage. It does not require any special or
unusually formatted diskettes. Any diskette
initialized for use with DOS 3.2 will do nicely.

FILER OPTIONS

        The DISK SLOT, DRIVE and VOLUME options specify
the device you will be using for storage. When the
system is booted, the default values are assigned.
You may change them in two ways. When instructed to
insert data diskette, you may specify the filer
options ,Ss,Dd,Vv (CONFIGURATION BOOT). You may also
change the options in the same fashion as any other
SUBSYSTEM option.

        No testing is done for validity so be sure the
options are correct before executing a FILER
command. Incorrect DRIVE or VOLUME values will cause
a recoverable system error. You will be notified and
receive control at the SUBSYSTEM MENU. Reenter the
FILER and correct the invalid option. A bad DISK
SLOT value is a little trickier. You will recieve
the same message and should follow the same
proceedure. After the SLOT value is corrected,
execute FILES DIRECTORY and the system will hang.
The system is waiting for the nonexistent drive to
shut off. You must press RESET. If your system has

the auto-boot monitor ROM, RESET will put you back
in the SUBSYSTEM MENU and everything should be
intact. If not you will be in the system monitor
with the * prompt. At this point type 3D0G then
RETURN to reenter the system. THAT'S THREE DEE ZERO
GO. 3D3G will put you in BASIC and destroy your
file.

FILES DIRECTORY

This command produces a list of MAGIC WINDOW
files on the current drive. Only files created by
the MAGIC WINDOW will be shown. They must be binary
or text files and have the filename suffix .MW to be
shown by the FILES DIRECTORY. The suffix does not
appear in the directory listing but will appear if
the diskette is catalogued from BASIC. The directory
listing appears at the bottom of the screen in two
columns as follows:

F003 TEST FILE   F099 BIG FILE
F003 FORMATTED   U002 UNFORMATTED

The F and U, followed by the length of the file
in sectors, indicates the file type. The directory
is listed eight files at a time. If more than eight
files exist, a flashing cursor will appear prompting
you to press a key to continue the listing. About
ten files can be shown before the first ones scroll
out of view. Pressing ESC will abort the listing.

FILENAMES

MAGIC WINDOW filenames must be 15 characters or
less in length starting with a alphabetic character.
They may include spaces, numbers and other special
characters. The suffix .MW is supplied automatically
by the system and should not be entered as part of
the filename. If you wish to load a text file
created by something other than the MAGIC WINDOW,
you must rename the file adding the .MW suffix
before the system will recognize it. The suffix was
included as insurance against loading invalid files.
You should not attempt to load fixed length text

files or binary files not created by the MAGIC
WINDOW. Entering a null string (just RETURN) will
cancel the operation in effect and return you to
function selection.

MAGIC WINDOW FILE TYPES

Two file types are supported for use with the
MAGIC WINDOW. They are FORMATTED (binary) and
UNFORMATTED (text) files. Both types offer storage
and retrieval of your text files. They differ in how
they are saved and the choice of type depends on
your application.

FORMATTED FILES

These files are binary images of your text file
as it is in memory. The entire file is treated as
one block of data. This blocking limits transfers to
the entire file. As well as text data, formatting
information is saved in the file. This additional
information includes the FORMAT SUBSYSTEM options,
the title line, tab stops and the line number of the
first non blank line of text in your file. As you
might guess, loading a formatted file recovers the
state of the system when the file was saved. This
makes it easy to interrupt your work then pick it up
later. Formatted file transfers are also faster than
unformatted transfers. For these reasons formatted
files are preferable for most work.

SAVING FORMATTED FILES

When executed, a directory of FORMATTED type
files will appear followed by the prompt ENTER
FILENAME. Enter the desired filename then press
RETURN. The entire file will be saved with format
information as prescribed for FORMATTED file types.
An updated directory listing will follow a
successful save operation.

LOADING FORMATTED FILES

When executed, a directory of FORMATTED type files will appear followed by the prompt ENTER FILENAME. Enter the desired filename to load then press RETURN. The file in memory will be replaced by the entire file from disk. The system format will be updated as described above under FORMATTED FILES.

DELETE FILE

When executed, a directory of all MAGIC WINDOW files is displayed and you are prompted for the filename you wish to delete from the disk. All deletions are final. Once deleted a file cannot be recovered.

CLEAR FILE IN MEMORY

When executed, the current file in memory is cleared. Once cleared the file in memory cannot be restored.

UNFORMATTED LOAD/SAVE

This function was saved for last because of its complexity. When executed, a new set of options and functions are displayed.

UNFORMATTED LOAD/SAVE

        TRANSFER ENTIRE FILE.......YES
        RANGE MARKED...............YES
        INCLUDE MARKERS............YES
        STARTING PARAGRAPH.........1
        NUMBER OF PARAGRAPHS.......1

        FILER SUBSYSTEM
        CHANGE RANGE OPTIONS
        SAVE FILE
        LOAD FILE
        INSERT FILE (AT CURSOR)

UNFORMATTED FILES

Text editing systems can be used for many functions other than creating corespondence. These include, but are not limited to, information collection, creating and editing source files for high level language interpreters, compilers, assemblers and so on. Lawyers and other professionals may require just a few paragraphs from a file and mixing portions of several files is sometimes useful. These applications do not require formatting information. In fact, contention could occur when dealing with more than one file. For these reasons, and others you may think of, the MAGIC WINDOW includes unformatted file management. Unformatted files are line blocked, variable length sequential access text files. Line length is limited to 80 characters for use with the editor. Each line is terminated with a Carriage Return (CTRL-M) and the end of file is marked with an End Of Text (CTRL-C) character. The MAGIC WINDOW will always add the end of file marker when saving an unformatted file but EOT is not required in a file to read it. If EOT is not present, the file will be read until DOS indicates an end of data error which is then handled by the error processing code. Unformatted files will assume the format the system is in when read. If the TEXT WIDTH is 40 characters, a file of 80 character lines will be broken between words to fit the system format. The opposite is not true. To pack a 40 character per line file into 80 character format, use the editor GLUE line command.

The UNFORMATTED LOAD/SAVE options specify if you wish to transfer a entire file and how you will specify the range of lines if only a portion is to be transferred. These options are bidirectional transfer options meaning they apply to files being loaded or saved.

TRANFER ENTIRE FILE

If YES the entire file will be transferred. This option always takes precedent over the other

options. If NO, a portion of the file is to be transferred. The method used to identify the range of lines is selected by the next option.

RANGE MARKED

There are two methods available to indicate the range to transfer. If this option is YES you will be prompted for the STARTING MARKER and ENDING MARKER. The system will search the file for the starting marker. When (and if) the starting marker is found, transfer will start and continue until the ending marker is found or the end of file is detected.

INCLUDE MARKERS

If indicated the markers will be included as the starting and ending lines of the data transferred.

MARKER SPECIFICATION.

A marker is a character or sequence of characters terminated by a carriage return. This is also the definition of a line of text stored on disk and of a line of text saved in text memory in compacted form (trailing spaces omitted). The markers (or lines) you enter will be compared to the lines in the file and must match in content up to (but not including) the carriage return specified in the marker line for the marker to be found. The comparison is case independant so upper and lower case are equivilant. Here are some sample hits and misses.

MARKER LINE: 'CHAPTER 1<CR>'

MARKER FOUND
FILE LINE: 'CHAPTER 1<CR>'
FILE LINE: 'Chapter 1<CR>'
FILE LINE: 'Chapter 1 - Getting Started<CR>'
MARKER NOT FOUND
FILE LINE: 'CHAPTER ONE<CR>'
FILE LINE: ' CHAPTER 1<CR>'

The apostrophe is included to delimit the lines and does not appear in the marker or file.

SPECIAL MARKERS

'<CR>' - A carriage return, or null line, marker will match any line in the file and is a convienient way to specify the first line of the file.

'<CTRL-C><CR>' - This is the file terminator that marks the end of a text file. Using this as the ending marker will specify a range marked by the end of file.

MARKER ERRORS

If one or both markers are not found, you will be notified. If the starting marker was not found, no transfer occured. If the ending marker was not found, the file was transfered to end of file.

STARTING PARAGRAPH & NUMBER OF PARAGRAPHS

If the RANGE MARKED option is NO, the range is identified numerically by paragraph number. The numbering starts with 1 and the maximum value is 254. Transfer starts at the STARTING PARAGRAPH number and continues for the NUMBER OF PARAGRAPHS desired.

PARAGRAPH SPECIFICATION

A paragraph is a line or lines delimited by indentation (leading spaces) or blank (null) lines. Here is a sample file.

Chapter 1   PARAGRAPH 1
     Any line indented from   PARAGRAPH 2
the left margin marks the
start of a paragraph.
     Paragraph.   PARAGRAPH 3
     Paragraph.   PARAGRAPH 4

Blank lines also indicate the   PARAGRAPH 5
end of the previous paragraph
and the next starts with the
next non blank line.

     As you can see, selecting a range of paragraphs
can be tricky business unless your file is
consistently structured. Usually markers are a more
obvious and less error prone way to identify a range
of text.

SAVE FILE

     When executed, this function will display the
directory of unformatted files and prompt you for a
filename. After entering the filename you will be
prompted for markers if the RANGE MARKED option is
in effect. The specified range of text is then saved
replacing any file with the same filename. To
facilitate garbage collection the file is deleted
before it is saved. For this reason you should
always save the file under a new filename then
delete the old file when the operation is
succesfully completed without marker errors or other
misfortunes. A successful save is followed by an
updated directory listing.

LOAD FILE

     When executed, this function displays the
directory of unformatted files then prompts for a
filename. After entering the filename you will be
prompted for markers if RANGE MARKED is in effect.
The specified range of text will be loaded from the
file replacing the file in memory. If a line is
longer than the TEXT WIDTH specified in the FORMAT

SUBSYSTEM it will be broken between words to fit the system format.

INSERT FILE (AT CURSOR)

When executed, the directory of unformatted files is displayed and you are prompted for a filename. After entering a filename you will be prompted for markers if RANGE MARKED is in effect. The specified range of text is then inserted in the file in memory immediately before the line the cursor was at when you left the EDITOR SUBSYSTEM. Appending is accomplished by moving the cursor to the end of the file then inserting a file. It is a good idea to save your file before doing a file insert. If an error occurs you can reload the original file and try again.

PRINTER SUBSYSTEM

      PRINT LOWER CASE...........YES
      PRINT ALL PAGES............YES
      STARTING PAGE #............1
      PAGE STOP..................NO
      MARK PERFORATION...........NO

      SUBSYSTEM MENU
      CHANGE PRINT OPTIONS
      PRINT HARDCOPY
      PRINT SOFTCOPY

The printed document is usually the final product of a word processing system. The PRINTER SUBSYSTEM will create the document on either hard or soft media. HARDCOPY refers to paper created by your printer and SOFTCOPY is a text file image of the hardcopy saved on diskette. Let's talk about the print options.

PRINT LOWER CASE

When selected, lower case characters will be printed as lower case. If your printer cannot print lower case or you wish to convert the lower case to

uppers for printing, Select NO. Only the printed
output will be converted to upper case. The text
file in memory remains unchanged.

PRINT ALL PAGES

Sometimes only a few pages from the file need
to be printed. If you select NO, you will be
prompted for the starting page number and the number
of pages to print before the file is printed. The
page range values are the ones used by the editor,
not the ones that appear in the title line of the
final copy. The first page of the file is always 1
as far as the editor is concerned. The range is
restricted to page boundaries and will always print
a full page.

STARTING PAGE NUMBER

This numeric option specifies the page number
that will appear in the title line (in place of #)
on the first page printed. If you have selected a
range of pages this number will appear on the
starting page of that range. As pages are printed,
the number is incremented. Because one byte numbers
are used in the MAGIC WINDOW, the highest page
number that can be printed is 255.

PAGE STOP

If you desire the printer to stop before each
page is printed, select YES. This is common when
printing on letterhead and any kind of single sheet
paper. Before every page the printer will stop and
you will be requested to POSITION DOCUMENT. You
should align the paper to top of form and then press
RETURN to print that page.

MARK PERFORATION

Some printers cannot use form perforated paper
or single sheets. This is typical of electrostatic
and thermal printers. These printers produce a roll
of paper. When selected, a perforation line will be

marked at each page boundary so you may cut the roll into sheets if you desire. Because this function adds one line more to the form length it should not be used with standard perforated forms.

DOCUMENT POSITIONING

As with the editor, line 1 character 1 is the top and left most position on the page. Top of form should be set on the first line of the page. Do not set it where you want the first line to be printed (unless you have selected a TOP MARGIN value of 0). the boundaries where the text will be printed are defined in the FORMAT SUBSYSTEM. To position the text, for centering on the page, simply change the TOP and LEFT MARGIN values.

PRINTING HARDCOPY

Before printing the file make sure the PRINTER SLOT and RAM PRINTER DRIVER options are set correctly in the CONFIGURATION SUBSYSTEM. Trying to print to a slot that does not contain a printer card will cause the system to 'hang', forcing you to press RESET. When everything is set up and your printer is on line select PRINT HARDCOPY then press return. You will be requested to position the document then press RETURN. The screen will go blank and the printer should be printing your file. If you desire to abort the print job before it finishes press ESC and the printer will skip to top of form then stop. After the file has been printed the PRINTER SUBSYSTEM will reappear on the screen awaiting your next command.

PRINTING SOFTCOPY

An exact image of the hardcopy will be printed into a sequential text file. You will be prompted for the filename. Physically, this file is identical to the UNFORMATTED files used by the FILER and can be transferred by the FILER. In content, it is exactly the same as hardcopy printed by your printer.

Softcopy can be used in many ways. Form letters can be created and specific portions, such as name and address, can be read from a disk file to replace field identification markers in the form letter. While this function is not a part of the MAGIC WINDOW it is not difficult and can be written in BASIC. For some ideas consult the APPENDIX section on USING TEXT FILES. Another use for softcopy is printer spooling, if not in the classical sense at least in a practical sense. A final copy of your document with correct page numbers, formatting et al can be saved at high speed to a text file. A less trained operator can then print your file when convenient without knowledge of formatting. You and your computer are free to do other things. See the APPENDIX for a sample program to print a softcopy file.

SYSTEM EXIT

The final function in the SUBSYSTEM MENU is EXIT TO BASIC. The function will place you in BASIC and there is no returning to the MAGIC WINDOW without booting the system diskette. Make sure you have saved your work before you exit.

HIT RESET?

It happens to all of us! If your machine is of the APPLE II+ variety, RESET will place you in the SUBSYSTEM MENU. If not you will be dumped into the APPLE monitor. From there, type 3D0G RETURN to reenter the SUBSYSTEM MENU. The extent of the damage caused by RESET depends on what the system was doing when it happened. If the system was waiting for you to type something, chances are good that the program and text will be intact. The worst can happen if RESET occurs when saving a file to disk. In any case, carefully check your data after a RESET. Backup is the best way to protect yourself from disaster. Every few pages, save your file.

BOOT CAMP

EDITING COMMANDS

EDITING COMMANDS

Editing commands are issued using the CTRL key on the Apple keyboard. When in edit mode, holding the CTRL key down and pressing any character key will cause a function to be executed. These functions are divided into groups performing cursor motion, character editing, line editing and other functions.

SUBSYSTEM MENU (EDITOR EXIT)

Command Key => SHIFT/CTRL-P or CTRL-@

To exit the editor at any time, hold both the CTRL and SHIFT keys down then press P. When using an Apple II+ or Apple II with auto boot ROM, pressing the RESET key has the same effect. You are cautioned against using the RESET key. If RESET is pressed while an editor function is being executed, you may butcher your text or lose it completely.

CURSOR MOTION GROUP

As you type, the cursor moves to the next position to accept the next character. To position the cursor without changing any characters, use one of the following cursor motion commands.

Command Key => CTRL-H or LEFT ARROW <--

Function: CURSOR LEFT ONE CHARACTER

The cursor moves one character to the left. If at the left margin it will not move. The cursor remains on the same line.

Command Key => CTRL-U or RIGHT ARROW -->

Funtion: CURSOR RIGHT ONE CHARACTER

The cursor moves one character to the right. If at the right margin it will not move. The cursor remains on the same line.

Command Key => CTRL-Q

Function: CURSOR UP ONE LINE

The cursor moves one line toward the top of the page. If moved into the top margin, it will continue to the last line of text on the preceeding page. The horizontal position of the cursor is not changed.

Command Key => CTRL-Z

Function: CURSOR DOWN ONE LINE

The cursor moves one line toward the bottom of the page. If moved into the bottom margin, it will continue to the first line of text on the following page. Horizontal position is not changed.

Command Key => CTRL-W

Function: CURSOR UP HALF WINDOW

The cursor moves 12 lines toward the top of the page. Margins are not counted as lines. If the top margin is reached, the cursor continues to the previous page. This function is used to scan backwards by half screens.

Command Key => CTRL-X

Function: CURSOR DOWN HALF WINDOW

The cursor moves 12 lines toward the bottom of the page. Margins are not counted. If the bottom margin is reached, the cursor advances to the next page. This function is used to scan forward by half screens.

Command Key => CTRL-E

Function: CURSOR TO TOP OF FILE

The cursor moves to the first line of text. The horizontal position is not changed. The page number

is also reset to one. For example, if you started
with an empty file, positioned the cursor to page 3
then typed a word, the first line of text would be
on page three preceeded by two blank pages. Invoking
the cursor to top of file command will ignore all
blank pages precedding the first line of text and
indicate the first line is on page one.

Command Key => CTRL-C

Funtion: CURSOR TO END OF FILE

        The cursor moves to the last line of text. The
horizontal position remains unchanged.

Command Key => CTRL-M or RETURN

Function: CARRIAGE RETURN & LINE FEED

        The cursor moves to the left margin and is
advanced one line toward the bottom of the page. If
DOUBLE SPACING is selected in the FORMAT SUBSYSTEM,
the cursor moves down two lines.

Command Key => CTRL-Y (menu selection)

Function: CLEAR ALL TAB STOPS

        The MAGIC WINDOW uses tab stops much like a
typewriter. The only difference is that you can tab
backwards as well as forward. This command clears
all existing tab stops. When the system is first
initialized all tabs will be cleared. Tabs are used
for cursor positioning in the editor and have no
effect when printing.

Command Key => SHIFT/CTRL-N or CTRL-^

Funtion: SET TAB STOP

        A tab stop is set at the current cursor
position. As well as indenting and typing tabular

information, tab stops are useful in reviewing text. Setting tab stops at about 20 characters from the margins makes reading easier.

Command Key => SHIFT/CTRL-M or CTRL-]

Function: CLEAR TAB STOP

The tab stop at the cursor position is cleared.

Command Key => CTRL-A

Function: TAB LEFT

The cursor moves to the next set tab stop to the left of the cursor. If there is no set tab stop to the left or if already at the left margin, the cursor does not move.

Command Key => CTRL-S

Function: TAB RIGHT

The cursor moves to the next set tab stop to the right of the cursor. If there are none or the cursor is at the right margin, it will not move.

Command Key => CTRL-Y (menu selection)

Function: ASSIGN SEARCH STRING

When editing large text files, it is often convenient to position the cursor at the beginning of a particular word or phrase. The CTRL-Y command will show you a menu of edit functions. Use the ARROW keys to select 'ASSIGN SEARCH STRING' then press RETURN. This command will prompt you to type the phrase you will be looking for. The phrase is entered without case selection. Case selection is not important since the search routine considers a capital equivalant to a lower case character. After typing the phrase press the RETURN key. All characters, including leading and trailing blanks are part of the assigned search string. Once

assigned, this string remains assigned until another
is entered. The maximum string length is one line as
defined by TEXT WIDTH in the FORMAT SUBSYSTEM.

Command Key => CTRL-N

Function: CURSOR TO NEXT OCCURANCE OF STRING

     This routine will position the cursor at the
first character of the next occurance of the
previously assigned search string. The search is
always in the forward direction, starting at the
current cursor position. If the string is not found,
the cursor will end up at the end of the file.

Command Key => CTRL-P

Function: ENTER CURSOR TO TOP OF PAGE SEQUENCE

     CTRL-P is the first character of a sequence
that causes the cursor to be homed to the first
character position on some page. Borrowing some
computerese, we can address the desired page in four
different ways or modes. They are implied, absolute,
implied relative and relative addressing. The
characters entered after CTRL-P define the desired
mode. Pressing RETURN implies the same page the
cursor is on. Typing a number followed by RETURN
absolutely defines the desired page. The left or
right ARROW followed by RETURN implies the previous
or following page relative to the current page. An
ARROW followed by a number and RETURN indicates the
direction and number of pages to move relative to
the current page.

PAGING SEQUENCES (<- ARROWS ->   n=number)

CTRL-P,RETURN   Cursor to top of page
CTRL-P,n,RETURN   Cursor to top of page n
CTRL-P,<-,RETURN   Cursor to top of previous page
CTRL-P,->,RETURN   Cursor to top of next page
CTRL-P,<-,n,RETURN Cursor backwards n pages
CTRL-P,->,n,RETURN Cursor forwards n pages

Because cursor positioning is so important to electronic text editing, the MAGIC WINDOW cursor motion commands occupy a large number of command keys. Key position was selected for convenience. The most common motions are grouped at the left side of the keyboard next to the CTRL key. Coincidentaly they form a C shape. The keys at the top of the C cause the cursor to move up in the file. The lower keys move the cursor down. The keys in the middle tab left and right. The exceptions to this rule are the RETURN, LEFT & RIGHT ARROWS and seldom used string search commands. The search commands are in the middle of the keyboard, out of the way. The others were selected because of habit and natural visual identification.

COMMON CURSOR MOTION PICTORIAL
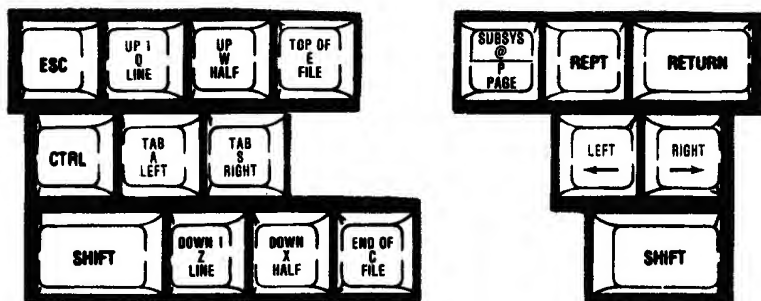
CHARACTER EDITING GROUP

This group of commands perform operations on the single character at the cursor location. There are four character editing commands.

Command Key => CTRL-D

Function: DELETE CHARACTER

The character at the cursor position is deleted from the file. All the characters on the line right of the cursor are moved one character to the left. Any character deleted in this manner cannot be recovered. The cursor position is not changed.

Command Key => CTRL-F

Function: INSERT BLANK CHARACTER

A space character is inserted at the cursor position. The characters at and to the right of the cursor are moved one space to the right to make room for the inserted space. The command is not performed if the insert would cause a character to be pushed beyond the right margin. This method is not generally effective for inserting more than a few characters. When a new phrase has to be added the suggested method is the 'cut and paste' operation performed with the SPLIT & GLUE operations. These are described in the next section on LINE EDITING COMMANDS.

Command Key => CTRL-V

Function: COPY CHARACTER (DITTO)

This command copies the character directly above the cursor into the cursor position then advances the cursor position.

Command Key => CTRL-B

Function: ENTER CONTROL CHARACTER

This command allows a control character to be placed in the file. When the ENTER CONTROL CHARACTER message is displayed, the next character typed will be converted to a control character then placed in the file at the cursor position. Control characters are displayed as flashing uppercase. These non-printable characters may be used by some printers for ribbon color or print size and many other control functions. All but one of the 32 control characters may be placed in the file. The illegal character is carriage return. It is reserved for the use of the editor and will not be accepted. Another one that is legal but should be avoided is the line feed or CTRL-J character. It will confuse the printer routines causing format problems. Also CTRL-D as the first character of a line will cause problems for the Disk Operating System resulting in system errors. CTRL-C marks the end of an unformatted file so don't put one in the middle of your text! In general make sure you know what you're doing when using this command. One more point of interest is that the MAGIC WINDOW considers these characters to occupy a space like any other character and will justify with that in mind.

LINE EDITING

This group of commands perform editing functions on parts of lines, entire lines and groups of lines. A powerful editing technique simliar to the familiar 'cut and paste' operation is provided with the SPLIT and GLUE line functions. Most of the line editing commands use a line 'stack' or buffer for temporary storage while editing. The stack is very important and deserves discussion before detailing the line editing commands.

DELETED LINE STACK

The deleted line stack is implemented as a 16 line Last-In First-Out buffer. Whenever a line is deleted, it is pushed onto this stack. The cursor can be moved and the line popped off the stack and reinserted into the file. In this fashion the stack can be used to move lines around in the file and is a built-in safety feature against accidental line deletion. It is a temporary storage area. You should not leave important text on the stack because it is used by many of the line editing commands. The stack can only hold 16 lines before it overflows destroying the first line. Why only 16 lines? Memory use is the reason. To hold 16 lines of 80 characters requires 1280 bytes of memory. The stack is not part of the text file and steals memory that could be used for your text. For this reason the stack size was set at 16 lines. It is not a serious limitation and simply requires subdivision when moving a large number of lines.

Command Key => CTRL-K

Function: DELETE (KILL) LINE

The line at the cursor is deleted and pushed onto the stack. All lines below the cursor are moved up to replaced the space occupied by the deleted line. The cursor position is not affected.

Command Key => CTRL-R

Function: RECOVER LINE

The top (last deleted) line on the stack is inserted into the file before the line at the cursor position. The line at the cursor and all following are moved down to make space for the recovered line. The cursor remains at the recovered line.

EDITING WITH DELETE AND RECOVER

    As well as removing lines from the file, the DELETE line command can be used to move lines of text around in the file. Groups up to 16 lines can be deleted, then the cursor moved to the desired line and the text popped off the stack and reinserted with the RECOVER line command. When moving text with the DELETE and RECOVER commands, other editing commands should be avoided while the text is on the stack. Using any commands other than cursor positioning may destroy some of the lines on the stack.

Command Key => CTRL-I

Function: INSERT LINE

    A blank line is inserted before the line at the cursor. The line at the cursor and all following are moved down to make space. Inserting a blank line does not affect the stack or cursor position.

Command Key => CTRL-O

Function: CLEAR LINE

    The entire line at the cursor location is pushed onto the stack and blanked in the file. It is similar to the DELETE function but clears the line instead of deleting it. No other lines are affected and the cursor remains at the blank line.

Command Key => CTRL-L

Function: CLEAR TO END OF LINE

    The characters at the cursor and to the right are cleared to blanks. They are not placed on the stack and cannot be recovered. No other lines are affected and the cursor position is not changed.

Command Key => CTRL-T

Function: SPLIT LINE

The line is split at the cursor position. The characters at the cursor and to the right are inserted in a new line, left justified, immediately below the current line. The cursor position is not affected.

Command Key => CTRL-G

Function: GLUE LINES

Words from the line below the cursor are glued (appended) to the end of the current line. Spaces are inserted between words. The words are moved until no more will fit or the line below the cursor is empty. When empty, the line below the cursor is deleted from the file. When no more words will fit on the current line the cursor is moved to the next line. The GLUE command will always leave the cursor at the left margin.

EDITING WITH SPLIT AND GLUE

Two of the most common editing tasks are rephrasing a sentence and inserting a word or phrase into a sentence. SPLIT and GLUE were designed to make these operations simple. Move the cursor to the place where you want to insert material. SPLIT the line then start typing the new material. When you reach the end of the line and overtype a new line is inserted automatically for you. In this fashion you can insert as much material as you desire. When finished entering text, use the GLUE command to append the split portion of the line and repack the following lines in the paragraph. Never GLUE the last line of a paragraph as this will append the next paragraph. When material is to be replaced, isolate it with the SPLIT command at the start and end of the phrase. Once seperated from other text, the lines with the old material can be deleted. Now insert the new material and repack with GLUE LINES.

Command Key => CTRL-J

Function: JUSTIFY LINE

    In keeping with the what you see design philosophy, line justification is done by the text editor. Pressing CTRL-J will place the editor in justify mode and display a choice of functions at the top of the screen. A flashing line alternates with the current line indicating justify mode. The next character pressed is interpreted as a justify command. If valid it is performed and the cursor moves to the next line awaiting another justification command.

Sub-command Key => C

Function: CENTER LINE

    The line is centered between the margins. Internal spacing is not changed. The cursor is positioned to the center and advanced to the next line.

Sub-command Key => L

Function: LEFT JUSTIFY LINE

    Any leading spaces are removed shifting the line to the left margin. Internal spacing is not affected. The cursor is positioned at the left margin and advanced to the next line.

Sub-command Key => R

Function: RIGHT JUSTIFY LINE

    Any trailing spaces are removed shifting the line to the right margin. Internal spacing is not affected. The cursor moves to the right margin and advances to the next line.

Sub-command Key => E

Function: EXPAND LINE (LEFT/RIGHT JUSTIFY)

Spaces are inserted into the line to expand it to the right margin. Leading spaces (indenting) are not affected. The expand command equally distributes the additional spaces to give a balanced appearence. The cursor moves to the right margin and advances to the next line.

Sub-command Key => P

Function: PACK LINE

This is the complement of the EXPAND function. It will convert two or more spaces to a single space. Leading spaces are not affected. The cursor is left at the right margin on the next line.

Sub-command Key => S

Function: SKIP LINE

The cursor is moved to the center of the next line without affecting the text. SKIP is used to skip a line of text without cancelling justify mode.

Sub-command Key => anything but C,L,R,E,P or S

Function: CANCEL JUSTIFY MODE

As long as valid justify commands are entered the system remains in justify mode. Any invalid character will cancel justify mode and return the standard cursor and position line.

OTHER FUNCTIONS

We have already introduced some of the functions grouped in the CTRL-Y menu. The CLEAR TABS and ASSIGN SEARCH STRING function have been discussed. There are two more functions in the menu of auxiliary functions.

Command Key => CTRL-Y (menu selection)

Function: ASSIGN TITLE LINE

     Each file is permitted one title line. This function assigns the contents of the line the cursor is on as the title line. It will prompt you for the line number to place the title on. The line number must be in the top or bottom margin of the page. If the title contains the special character # the PRINTER SUBSYSTEM will print the current page number in place of #. The title will appear on the same line on every page. After the title is assigned the current line is cleared to blanks. You may use more than one # to reserve space for page numbers. The sequence ## reserves two spaces for one number. The sequence # # will print two numbers seperated by a space.

Command Key => CTRL-Y (menu selection)

Function: EXCHANGE KEY SETS

     One problem with using the APPLE keyboard for text editing is its incomplete set of available characters. Three characters can be displayed but cannot be directly generated from the keyboard. To overcome this limitation we swap between two sets of three keys. In this fashion we place the character from the current set into the file. Once in the file the character does not change when we swap sets again.

KEYBOARD KEYS ->	SHIFT-N	SHIFT-M	/
UPPER CASE SET 1	^	]	/
UPPER CASE SET 2		[	\
lower case set 1	~	}	/
lower case set 2	DEL	{	\

APPENDICES

APPENDIX   - 46 -

APPENDIX A - INSTALLING RAM PRINTER DRIVERS

To operate the MAGIC WINDOW with a RAM resident output routine a binary file named PRINTER is required on the data diskette. This file must be loaded into the system by the LOAD PRINTER DRIVER function in the CONFIGURATION SUBSYSTEM. This appendix outlines the construction and specifications of the file PRINTER. It is intended for the experienced APPLE assembly language programmer.

DRIVER LOCATION

Five hundred and twelve bytes of RAM have been reserved for custom interface. This continuous space begins at hex location $800. Two pages should accomodate a driver of moderate complexity and its data storage. Zero page locations $00 to $1F are available for use. Additional data storage at $300 to $3CF has been set aside by APPLE convention for custom I/O. These are guidelines only. Where the printer driver resides is totally up to you. When the file PRINTER is loaded it is located in the same space as when it was saved. SEE DISCUSSION OF BLOAD COMMAND IN THE APPLE DOS MANUAL. The MAGIC WINDOW extracts this information from DOS and uses the loading address to call the routine and print a character. This means that the first byte of your file must be the start of your output driver or jump to same.

REGISTER USAGE

The 6502 accumulator, X and Y index registers should be restored before exit to the values held on entry. Since exit is via RTS the stack must be preserved for it to work at all. The status register may be clobbered.

```
*
* RAM PRINTER DRIVER
* SPECIAL INITIALIZATION
* SAMPLE CODE
*
CSWL   EQU $36 ; MONITOR OUTPUT
CSWH   EQU $37 ; VECTOR
PRSLOT EQU $C100 ; SLOT 1
* CODE STARTS AT $300
       ORG $300
       OBJ $300
* PRESERVE REGISTERS
INIT   PHA
       TXA
       PHA
       TYA
       PHA
* SET SLOT 1 FOR OUTPUT
       LDA #PRSLOT
       STA CSWL
       LDA #>PRSLOT
       STA CSWH
* PRINT INITIALIZATION STRING
       LDY #0
PRTLUP LDA STRING,Y
       BEQ ALDONE
       JSR COUT
       INY
       BNE PRTLUP
* RESTORE REGISTERS AND RETURN
ALDONE PLA
       TAY
       PLA
       TAX
       PLA
* SEND ACCUM TO PRINTER ROM
COUT   JMP (CSWL)
* TEXT TO INITIALIZE PRINTER
STRING ASC 'SPECIAL CHARACTERS'
       BRK
```

APPENDIX B - USING SOFTCOPY

Softcopy is a exact replica of the printed
hardcopy. The storage media is a text file on
diskette. The task of converting softcopy to printed
hardcopy is simply reading a line from the file and
sending it to the printer. An APPLESOFT BASIC
program listing is provided to do this job.

Although simple, reading files of text material
can be tricky from BASIC. The biggest problem is
with punctuation and the BASIC INPUT statement.
BASIC will not allow some characters such as comma
and semicolon to be entered as input data. To
circumvent this snag two methods are available. One
is GETting the line character by character and
building a string. BASIC may be slower than your
printer when manipulating characters. The approach
taken in the sample program uses a small machine
language routine to read in a line and assign it to
a BASIC string. The routine is POKEd into memory
from data statements. It requires the input string
varaible to be the first defined variable. Because
the string is located in the input buffer it is
temporary only. Once read it should be assigned to
another string.

A decision is needed at the end of a page. For
single sheets we must stop the printing. How does
the BASIC program know when we are at the end of the
page? One easy answer is to MARK PERFORATION when
creating the softcopy. The sample program assumes
you have done this and provides complete
paging options.

```
1 REM   SOFTCOPY TO HARDCOPY
2 REM   SAMPLE PROGRAM
12 REM *
13 REM *   POKE IN INPUT ROUTINE
14 REM *
15 A$ = "A$ MUST BE FIRST VARIABLE DEFINED"
16 RESTORE :ADDR = 768
17 READ CODE: IF CODE < 0 THEN 50
18 POKE ADDR,CODE:ADDR = ADDR + 1: GOTO 17
19 REM
20 DATA   162,0 , 32,12,253 , 41,127 , 157,0,2
21 DATA   201,13 , 240,3 , 232 , 208,241 , 160,2
22 DATA   138 , 145,105 , 96 , -1
50 REM *
55 REM *   CONSTANTS
60 REM *
65 D$ =   CHR$ (4)
70 DIM P$(255):P$ = ""
75 FOR I = 1 TO 25:P$ = P$ + "----------": NEXT
80 ESC = 155
85 KB = 49152
90 KS = KB + 16
95 CEOL = 64668
100 REM *
105 REM *   SET UP JOB
110 REM *
115 TEXT : HOME : PRINT D$;"PR#";0
120 LIST ,10: POKE 34, PEEK (37)
125 INPUT "PRINTER CARD SLOT ?";SL
130 SL =   INT (SL): IF SL < 0 OR SL > 15 THEN 125
135 INPUT "INSERT DATA DISKETTE...PRESS RETURN";A$
140 PRINT D$;"CATALOG": PRINT
145 INPUT "TEXT FILE TO PRINT ?";FL$
150 INPUT "STOP AT PAGE MARK (Y/N) ?";A$
155 WHOA = 0: IF A$ < > "N" THEN WHOA = 1
160 INPUT "PRINT MARK (IF ANY) ''   ?";A$
165 MARK = 0: IF A$ < > "N" THEN MARK = 1
200 REM *
205 REM *   PRINT FILE
210 REM *
215 PRINT D$;"UNLOCK";FL$
220 PRINT D$;"OPEN";FL$
225 PRINT D$;"PR#";0: HOME
230 PRINT "POSITION DOCUMENT THEN PRESS RETURN"
```
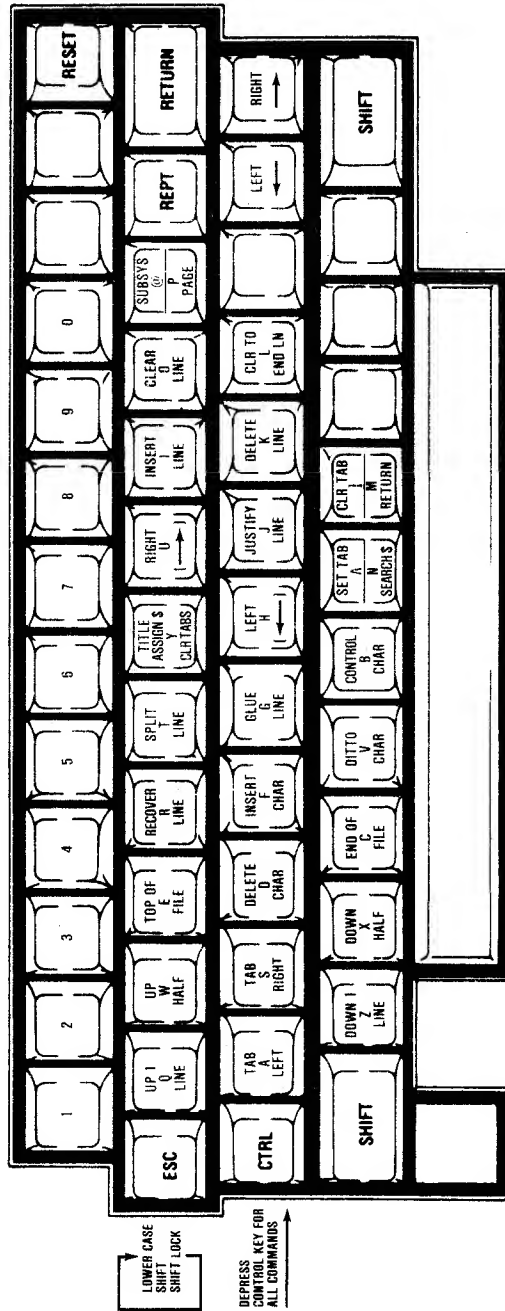
```
235 IF  PEEK ( - 16384) <  > 141 THEN 235
240 POKE  - 16368,0: PRINT D$;"PR#";SL
245 GOSUB 1000: REM    READ LINE
250 IF L$ = "" THEN  PRINT : GOTO 245
255 IF L$ =  CHR$ (3) THEN 300
260 IF NOT MARK AND L$ = MID$(P$,1,LEN(L$)) THEN 275
265 IF  PEEK (KB) = ESC THEN  POKE KS,0: GOTO 300
270 PRINT L$
275 IF WHOA AND L$ = MID$(P$,1,LEN(L$)) THEN 225
280 GOTO 245
300 REM *
305 REM *  END OF JOB
310 REM *
315 PRINT D$;"CLOSE";FL$: PRINT D$;"PR#";0: HOME
320 INPUT "ARE YOU FINISHED (Y/N) ?";A$
325 IF A$ = "N" THEN 135
330 TEXT : HOME : END
1000 REM *
1005 REM *  READ LINE FROM FILE
1010 REM *
1015 PRINT D$;"READ";FL$
1020 ADDR =  PEEK (105) + 256 *  PEEK (106) + 3
1025 POKE ADDR,0: POKE ADDR + 1,2: CALL 768
1030 L$ = A$: RETURN
```

APPENDIX   — 52 —

APPENDIX C -- EDITING COMMAND REFERENCE CARD

Keys (top to bottom / left to right):

RESET · RETURN · RIGHT → · SHIFT

REPT · LEFT ←

SUBSYS @ / P PAGE

CLEAR 0 LINE · CLR TO L END LN

INSERT I LINE · DELETE K LINE

RIGHT U (→) · JUSTIFY J LINE · CLR TAB M RETURN

TITLE ASSIGN $ Y CLR TABS · LEFT H (←) · SET TAB N SEARCH $

SPLIT T LINE · GLUE G LINE · CONTROL B CHAR

RECOVER R LINE · INSERT F CHAR · DITTO V CHAR

TOP OF E FILE · DELETE D CHAR · END OF C FILE

UP W HALF · TAB S RIGHT · DOWN X HALF

UP Q LINE · TAB A LEFT · DOWN Z LINE

ESC · CTRL · SHIFT

1 2 3 4 5 6 7 8 9 0

LOWER CASE SHIFT
SHIFT LOCK

DEPRESS CONTROL KEY FOR ALL COMMANDS ↑

APPENDIX

APPENDIX D - APPLE PARALLEL PRINTER INTERFACE

80 COLUMN LINE INITIALIZATION

    When using the APPLE PARALLEL interface with
the MAGIC WINDOW, some special initialization is
required. The interface uses a default line length
of 40 characters. To obtain the 80 character line
length required by the MAGIC WINDOW, the string of
characters "<CTRL-I>80N" must be sent to the
interface before the characters to be printed. This
may be accomplished by inserting them in the text
file before your subject matter. There is a better
way however.

    Using the RAM PRINTER DRIVER function, a small
assembly language routine can print these special
characters to the interface without messing up your
text file. The attatched listing shows the routine.
It must be entered into memory and BSAVEd on your
data disks under the filename PRINTER. Here is the
way to do so.

    enter the monitor from BASIC

>CALL -151 <RETURN>

    enter the following hex codes

*800:48 98 48 A9 00 85 36 AD <RETURN>
*808:C5 0A 09 C0 85 37 A0 00 <RETURN>
*810:B9 21 08 F0 06 20 1E 08 <RETURN>
*818:C8 D0 F5 68 A8 68 6C 36 <RETURN>
*820:00 89 B8 B0 CE 00 <RETURN>

    return to BASIC via DOS

*3D0G <RETURN>

    insert your data diskette and save it

>BSAVE PRINTER,A$800,L$80

Now you can try it out. Boot the MAGIC WINDOW then insert your data diskette with the file PRINTER on it. Go to the CONFIGURATION SUBSYSTEM and LOAD RAM PRINTER DRIVER. After a few seconds the file will be loaded and the options should indicate RAM PRINTER DRIVER..........YES. Make sure the PRINTER SLOT option identifies the slot containing the APPLE PARALLEL interface because the PRINTER code uses this value to communicate with the interface. Okay, type some text and print it out. If it works (you've entered the code correctly) UPDATE the Configuration boot file And your 40 column blues are over forever!

APPENDIX E - MAKING A DATA DISKETTE

     To save wear and tear on your MAGIC WINDOW
diskette we suggest you make a DATA diskette. The
DATA diskette will give you about 100K of memory for
your MAGIC WINDOW files. This can range up to 40
pages depending upon the FORMAT of your page.
The procedure should be followed for each diskette
that you will need.

     To make a DATA diskette you will need an APPLE
initialized diskette. The procedure to initialize a
disk is outlined in the DOS manual.

     Once you have an initialized diskette you
should boot up the MAGIC WINDOW. First you must get
into the CONFIGURATION SUBSYSTEM. There are several
options available to you, one of which is the UPDATE
BOOT CONFIGURATION FILE. When it comes time to make
the DATA diskette this is the option you will use.

     Before making the DATA diskette you should make
sure that it will be configured as you like. At the
top of the CONFIGURATION SUBSYSTEM menu there are
seven options. Make sure that these options are the
way that you want them to appear on your DATA
diskette. If you need to make any changes then refer
to PAGE 9.

     Once all these options are correct for your
system then insert the DATA diskette and position
the inverse bar over UPDATE BOOT CONFIGURATION FILE.
Press return and the disk drive will whir and click
as it normally does and then come to rest. Your
diskette is now prepared to accept MAGIC WINDOW
files.

     You can change the CONFIGURATION FILE and save
it to your DATA diskette without harming the MAGIC
WINDOW files.